



Satisfying ternary permutation constraints by multiple linear orders or phylogenetic trees



Leo van Iersel^a, Steven Kelk^b, Nela Lekić^b, Simone Linz^c

^a Delft Institute of Applied Mathematics, Technische Universiteit Delft, Mekelweg 4, 2628CD, Delft, The Netherlands

^b Department of Knowledge Engineering (DKE), Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands

^c Department of Computer Science, The University of Auckland, Auckland, New Zealand

ARTICLE INFO

Article history:

Received 15 October 2014

Received in revised form 23 April 2015

Accepted 7 June 2015

Available online 11 June 2015

Communicated by R. Giancarlo

Keywords:

Permutation constraint satisfaction problem

Linear orders

Phylogenetic trees

ABSTRACT

A ternary permutation constraint satisfaction problem (CSP) is specified by a subset Π of the symmetric group S_3 . An instance of such a problem consists of a set of variables V and a set of constraints \mathcal{C} , where each constraint is an ordered triple of distinct elements from V . The goal is to construct a linear ordering α on V such that, for each constraint $(a, b, c) \in \mathcal{C}$, the ordering of a, b, c induced by α is in Π . Excluding symmetries and trivial cases there are eleven such problems, and their complexity is well known. Here we consider the variant of the problem, denoted $2\text{-}\Pi$, where we are allowed to construct two linear orders α and β and each constraint needs to be satisfied by at least one of the two. We give a full complexity classification of all eleven $2\text{-}\Pi$ problems, observing that in the switch from one to two linear orders the complexity landscape changes quite abruptly and that hardness proofs become rather intricate. To establish the proofs, we use several computer-aided searches. Each such search returns a small instance with a unique solution for a given problem. We then focus on one of the eleven problems in particular, which is closely related to the $2\text{-CATERPILLAR COMPATIBILITY}$ problem in the phylogenetics literature. We show that this particular CSP remains hard on three linear orders, and also in the biologically relevant case when we swap three linear orders for three phylogenetic trees, yielding the $3\text{-TREE COMPATIBILITY}$ problem. Lastly, we give extremal results concerning the minimum number of trees required, in the worst case, to satisfy a set of rooted triplet constraints on n leaf labels.

© 2015 Published by Elsevier B.V.

1. Introduction

A *ternary permutation constraint satisfaction problem* (CSP), sometimes also known as an *ordering CSP*, is specified by a subset Π of the symmetric group S_3 . An instance of such a problem consists of a set of variables V and a set of constraints \mathcal{C} , where each constraint is an ordered triple of distinct elements from V . The goal is to construct a linear order α on V such that, for each constraint $(a, b, c) \in \mathcal{C}$, the ordering of a, b, c induced by α is in Π . For example, if $\Pi = \{123, 132\}$ then for each constraint (a, b, c) we require that $\alpha(a) < \alpha(b) < \alpha(c)$ or $\alpha(a) < \alpha(c) < \alpha(b)$, which can be summarized as $\alpha(a) < \min(\alpha(b), \alpha(c))$. Excluding symmetries and trivial cases there are eleven such problems, some of which have ac-

E-mail addresses: l.j.v.iersel@gmail.com (L. van Iersel), steven.kelk@maastrichtuniversity.nl (S. Kelk), nela.lekic@maastrichtuniversity.nl (N. Lekić), s.linz@auckland.ac.nz (S. Linz).

<http://dx.doi.org/10.1016/j.tcs.2015.06.021>

0304-3975/© 2015 Published by Elsevier B.V.

Table 1

The complexity of the 11 ternary permutations CSPs, in the case of 1 linear order (1LO) (see [13]) and 2 linear orders (2LO) (this article).

		1LO	2LO
Π_0 (linear ordering)	123	P	NPC
Π_1	123, 132	P	NPC
Π_2	123, 213, 231	P	P
Π_3	123, 231, 312, 321	P	P
Π_4	123, 231	NPC	NPC
Π_5 (betweenness)	123, 321	NPC	NPC
Π_6	123, 132, 231	NPC	NPC
Π_7 (circular ordering)	123, 231, 312	NPC	P
Π_8	$S_3 \setminus 123, 231$	NPC	P
Π_9 (non-betweenness)	$S_3 \setminus 123, 321$	NPC	NPC
Π_{10}	$S_3 \setminus 123$	NPC	P

quired specific names in the literature, such as *betweenness* [7] and *cyclic ordering* [9]. A full complexity classification of the 11 problems is given in [13] and summarized in Table 1. Due to their fundamental character these problems have stimulated quite some interest from the approximation [11], parameterized complexity [12] and algebra [2] communities.

In this article we consider the variant of the problem, denoted k - Π , where we are allowed to construct k linear orderings and each constraint needs to be satisfied by at least one of them. In Section 3, we give a full complexity classification of all eleven 2 - Π problems, observing that in the switch from one to two linear orders the complexity landscape does not behave monotonically (see Table 1). We note that for a single linear ordering the polynomial-time variants can be solved with variations of topological sorting, and the hard variants can be proven NP-complete using fairly straightforward reductions [13]. In the case of two linear orderings all the polynomial-time variants are *trivially* solvable while, for the other variants, establishing the NP-completeness is much more challenging, requiring a wide array of novel gadgets and constructions.

Our motivation for studying the k - Π problems is that, in practice, constraints may come from different sources, and it might not be known which constraints come from a common source. In such situations, it makes sense to construct different solutions that commonly satisfy all given constraints. A good example of a practical problem where this phenomenon occurs can be found in *phylogenetics*, a branch of computational biology concerned with the inference of evolutionary histories [23]. We discuss this phylogenetics problem in Sections 5–8.

Here we are given a set of *rooted triplets* which are leaf labeled, rooted binary trees on three leaves. Rooted triplets have a central and recurring role within phylogenetics due to the fact that they can be viewed as the atomic building blocks of larger evolutionary histories [23]. Determining whether all elements in a set \mathcal{R} of rooted triplets can simultaneously be embedded into a tree-like hypothesis of evolution, known as a phylogenetic tree, can be computed in polynomial-time by using the classical algorithm of Aho et al. [1]. If such a phylogenetic tree exists, then \mathcal{R} is called *compatible*. Frequently however such a tree does not exist and one might be interested in finding a maximum-size compatible subset of \mathcal{R} . This problem was first shown to be NP-hard by Bryant [4] as well as later again by [15,25], and several exact and approximation algorithms have been developed [6,25].

Recently, the impact of reticulation events, like hybridization and horizontal gene transfer, on the evolutionary history of certain groups of species is widely acknowledged. In the presence of reticulation, different parts of a genome often contain different tree-like evolutionary signals that need to be untangled [14,18,19]. To do this, a possible approach is the following. Instead of constructing one phylogenetic tree that embeds all triplets of a maximum-size compatible subset S of \mathcal{R} and ignoring all evolutionary information of the triplets that are in \mathcal{R} but not in S , it seems sensible to ask for the smallest integer k such that \mathcal{R} can be partitioned into at most k compatible subsets of triplets. For each such subset, one can use the algorithm of Aho et al. to find a phylogenetic tree, which represents the evolutionary history of a part of the genome. The collection of all these phylogenetic trees can also be used as input to methods that construct a leaf-labeled acyclic digraph (called a “rooted phylogenetic network”) that contains all the computed phylogenetic trees and hence represents the evolution of the whole genome, see e.g. [14]. This approach was introduced in [16] and the associated decision problem is called k -TREE COMPATIBILITY. In their paper, the authors show that 2 -TREE COMPATIBILITY is NP-complete. We observe that the problem 2 - Π_1 (which corresponds to the $\Pi = \{123, 132\}$ example given earlier) is equivalent to the problem studied in [16], with one extra restriction: the two phylogenetic trees we construct must be “caterpillars”, yielding the 2 -CATERPILLAR COMPATIBILITY problem. Our hardness result for 2 - Π_1 thus supplements the hardness result of Linz et al. (and, as a spin-off result presented in Section 4, establishes a link to the literature on the dichromatic number problem [3]). To further extend this result we show in Section 5 that 3 - Π_1 is hard and, building on this machinery, 3 -TREE COMPATIBILITY is also hard. As with many of the hardness results in this article we make heavy use of special gadgets that are unique solutions to the set of constraints that they imply. Such uniqueness gadgets could be of independent interest. To find these gadgets, we use several computer-aided searches that are described in Section 6. Each such search returns a small instance of one of the eleven k - Π problems and provides a “backbone structure” for the construction used in the associated proof showing that the problem is NP-complete. Like other proofs that were established with the assistance of a computer (e.g. [8,22]), our results emphasize the potential importance of computers in establishing hardness results.

Download English Version:

<https://daneshyari.com/en/article/435552>

Download Persian Version:

<https://daneshyari.com/article/435552>

[Daneshyari.com](https://daneshyari.com)